

Programmation Web 1

GI1

R.J.

Printemps 2024



2 HTML (suite)

Les frames

- Grâce à la technique des "frames" ("cadres" en français), il est possible d'afficher plusieurs pages HTML dans différentes zones, i.e., cadres ou parties, d'un même onglet d'un navigateur Web.
- Pour ce faire, il faut un fichier HTML qui définira l'agencement des cadres ; c'est un fichier qui a pour particularité d'avoir les balises "body" remplacées par les balises spéciales "frameset".
Il faudra aussi autant de fichiers HTML "contenus" que de cadres ; dans chaque cadre sera chargé un de ces fichiers là.
- Donc pour une page contenant N frames, il faut $N + 1$ fichiers.

Remarque : Dans une utilisation normale, on chargera N fichiers HTML différents (N contenus différents) ; dans ce cas là, on a donc les $N + 1$ fichiers. Mais il se pourrait qu'on désire charger un même contenu plus qu'une fois, et dans ce cas là, il y aura bien sûr moins que $N + 1$ fichiers.

- Comme un tableau, le fichier qui définit l'agencement via la balise `<frameset>` se construit en lisant les cadres de gauche à droite et de haut en bas.
- Commençons par un premier exemple :

-- **Créons un fichier "mainpage.html" qui contient :**

```
<!DOCTYPE html>
<html>
  <head>
    <title> mon site </title>
  </head>
  <frameset cols="350,*">
    <frame src="frame1.html" name="colgauche"/>
    <frame src="frame2.html" name="coldroite"/>
  </frameset>
  Attention : Ce site a été construit avec des frames.
  Il semble que votre navigateur ne supporte pas ces balises.
  </frameset>
</html>
```

Notez bien qu'il n'y a pas les balises "body" (elles ont été remplacées par les balises "frameset").

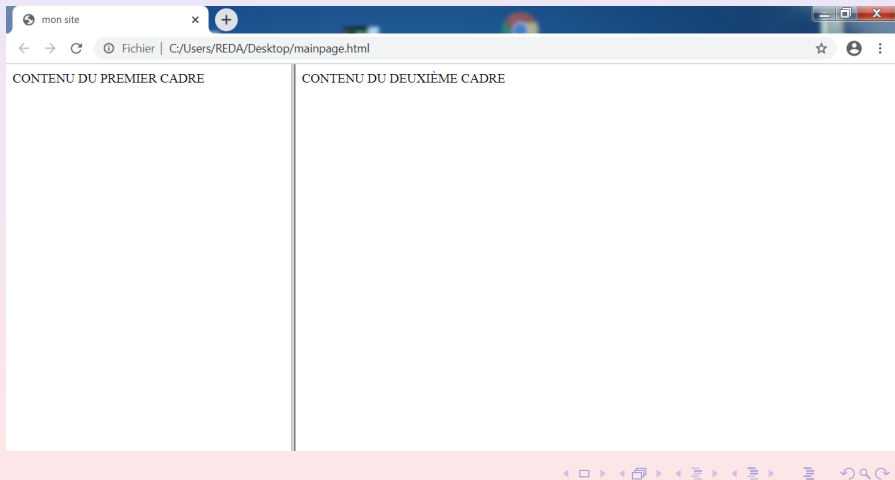
-- **Créons un fichier "frame1.html" qui contient :**

```
<!DOCTYPE html>
<html>
  <head>
    <title> TITRE </title>
  </head>
  <body>
    CONTENU DU PREMIER CADRE
  </body>
</html>
```

-- **Créons un fichier "frame2.html" qui contient :**

```
<!DOCTYPE html>
<html>
  <head>
    <title> TITRE </title>
  </head>
  <body>
    CONTENU DU DEUXIÈME CADRE
  </body>
</html>
```

— — **Quand on ouvre le fichier "mainpage.html" avec le navigateur on obtient :**



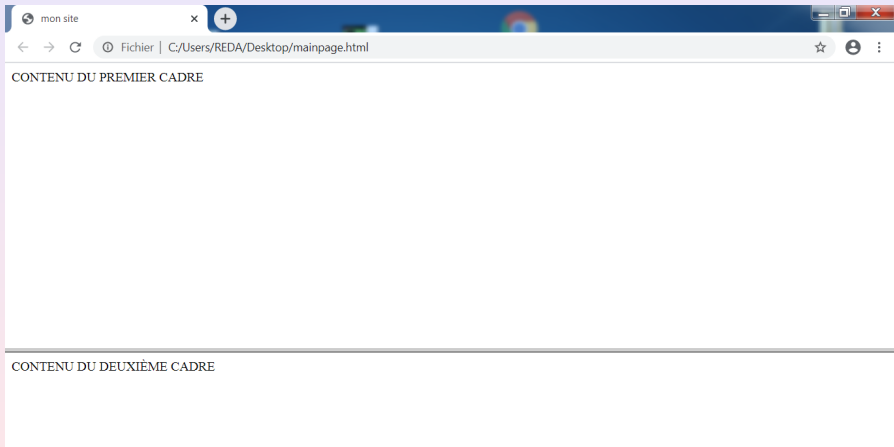
Dans le précédent exemple, on a créé donc une page Web à 2 frames qui sont disposées en colonnes.

La première fait 350 pixels de large et se trouve à gauche de la fenêtre, alors que la seconde occupe le reste disponible de la fenêtre du navigateur (et se trouve donc à droite).

On a appelé la colonne de gauche "colgauche", dans laquelle on a chargé le fichier "frame1.html". Et on a appelé la colonne de droite "col-droite", dans laquelle on a chargé le fichier "frame2.html".

Au cas où le navigateur Web ne supporterait pas les balises de manipulation des frames, on affichera un message indicatif via les balises "noframes".

- Si on remplace dans le fichier "mainpage.html" l'attribut *cols* de la balise `<frameset>` par un attribut *rows*, on aura :



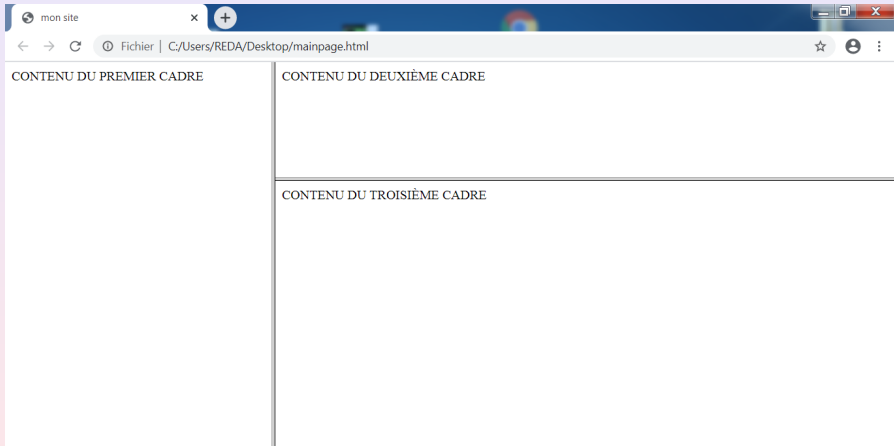
- La balise `<frameset>` admet comme couples attribut valeur :
 - `cols="pixels ou %, pixels ou % ..."` : crée des cadres verticaux (colonnes), dont la largeur peut être spécifiée en pixels, ou en pourcentage de la fenêtre totale.
Il est possible d'utiliser le caractère `"*"` pour indiquer que la colonne occupera l'espace restant.
Les largeurs sont séparées par des virgules,
 - `rows="pixels ou %, pixels ou % ..."` : crée des cadres horizontaux (lignes), dont la hauteur peut être spécifiée en pixels, ou en pourcentage de la fenêtre totale.
Il est possible d'utiliser le caractère `"*"` pour indiquer que la ligne occupera l'espace restant.
Les hauteurs sont séparées par des virgules,
 - `bordercolor="couleur"` : adopte la couleur donnée en argument, comme étant la couleur de la bordure des cadres,
 - `border="N"` : Met à N, l'épaisseur de la bordure entre les cadres.
Notez bien que si on veut que les cadres n'aient pas de bordures, il faudra dans ce cas là mettre `border="0"`.

- La balise `<frame />` admet comme couples attribut valeur :
 - `src="fichier-html"` : indique le fichier HTML chargé dans le cadre,
 - `name="nom"` : attribue un nom au cadre,
 - `marginheight="N"` : définit une marge de N pixels en haut et en bas du cadre (par rapport au contenu),
 - `marginwidth="N"` : définit une marge de N pixels du côté gauche et du côté droit du cadre (par rapport au contenu),
 - `scrolling="yes"` ou `scrolling="no"` : autorise ou non la présence de la barre de défilement (ascenseur de défilement) dans le cadre.



- Exemple d'une utilisation conjointe des attributs *cols* et *rows* :

```
<!DOCTYPE html>
<html>
  <head>
    <title> mon site </title>
  </head>
  <frameset cols="30%,70%">
    <frame src="frame1.html"/>
    <frameset rows="30%,70%">
      <frame src="frame2.html"/>
      <frame src="frame3.html"/>
    </frameset>
  </frameset>
</html>
```



Les formulaires

Préambule

- Un formulaire permet à des visiteurs de saisir et de transmettre des informations. Avec les formulaires, le langage HTML ouvre la porte à l'interactivité via des éléments interactifs.
- Un visiteur du site Web pourra par exemple saisir du texte dans une zone de texte ou cocher une case à cocher ...
- Un formulaire est déclaré grâce aux balises `<form>` `</form>`.
- On utilise généralement 3 attributs de la balise `<form>` :
 - *name* qui indique le nom que l'on souhaite donner au formulaire,
 - *action* qui indique l'URL où vont être envoyées les données soumises depuis le formulaire. Généralement, il s'agit d'une page Web "dynamique" (ayant l'extension `.php` ou `.asp`) qui va sauvegarder des données dans une base de données et/ou faire des traitements spécifiques sur des données,
 - *method* qui indique la méthode d'envoi des données, qui est soit "get" ou soit "post".

Préambule

- Exemple :

```
<form      name="mon-formulaire"      action="page-reception.php"  
method="get">
```

...

```
</form>
```

- On va insérer les éléments interactifs (qui peuvent être de différents types) entre les balises `<form>` et `</form>`.

Les méthodes get et post

- Un formulaire peut envoyer les informations d'une page Web à une autre page, de deux manières différentes :
 - – Avec la méthode get, les données du formulaire seront transmises en complément de l'URL de la page Web ciblée ; on verra une/des variable(s) et une/des valeur(s) s'ajouter à la fin de l'URL dans la barre d'adresse,
 - – avec la méthode post, les données du formulaire seront transmises d'une façon cachée ; les variables seront alors invisibles pour l'utilisateur.
- Exemple d'envoi d'une variable appelée "lieu" qui vaut "ENSATE" à une page Web `http://www.example.com/page-reception.php` :
 - – Avec la méthode get, l'URL sera modifié en `http://www.example.com/page-reception.php?lieu=ENSATE`
 - – avec la méthode post, l'URL ne sera pas modifié.

Ajout d'un champ texte sur une seule ligne (champ texte simple)

- Nous allons utiliser la balise `<input />` avec le couple attribut valeur `type="text"`.
- L'attribut *name* de la balise `<input />` permet de spécifier le nom de la variable qui sera envoyé.
- L'attribut *value* de la balise `<input />` permet de donner une valeur par défaut à la variable, i.e., le champ sera affiché pré-rempli.
- Exemple :

Entrez votre login : `
`

`<input type="text" name="login" value=".gi1"/>`

Ajout d'un champ de saisie/sélection

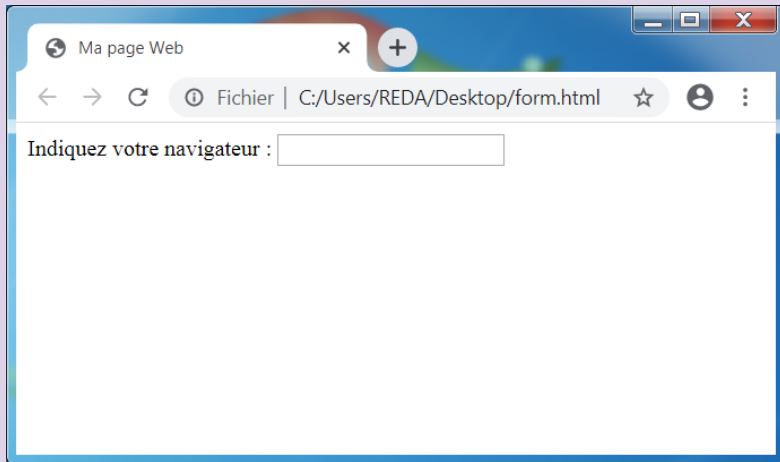
- C'est un champ qui permet de :
 - – Sélectionner une valeur parmi une liste prédéfinie de valeurs,
 - – sélectionner une valeur parmi une liste restreinte de valeurs ; la sélection portera sur uniquement les valeurs qui comportent un caractère ou une chaîne de caractères saisi(e) dans le champ par le visiteur,
 - – saisir une nouvelle valeur.
- Exemple :

Indiquez votre navigateur :

```
<input list="liste-navigateurs" name="navigateur"/>
<datalist id="liste-navigateurs">
    <option value="Internet Explorer"></option>
    <option value="Google Chrome"></option>
    <option value="Mozilla Firefox"></option>
    <option value="Safari"></option>
    <option value="Netscape Navigator"></option>
</datalist>
```

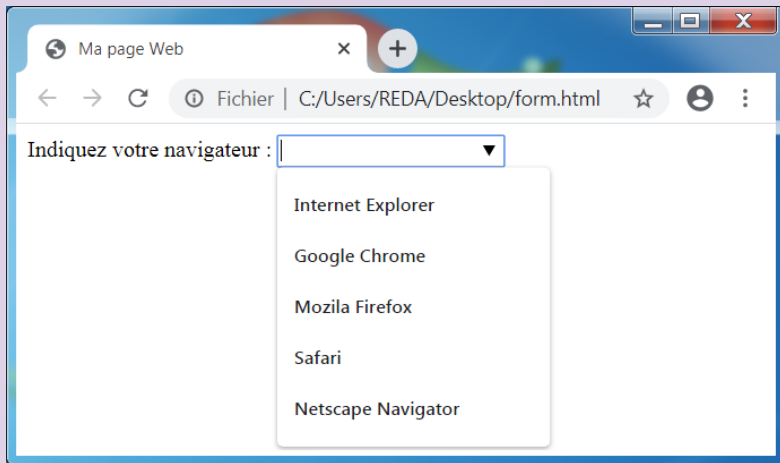
Ajout d'un champ de saisie/sélection

Quand on ouvre le fichier HTML on obtient :



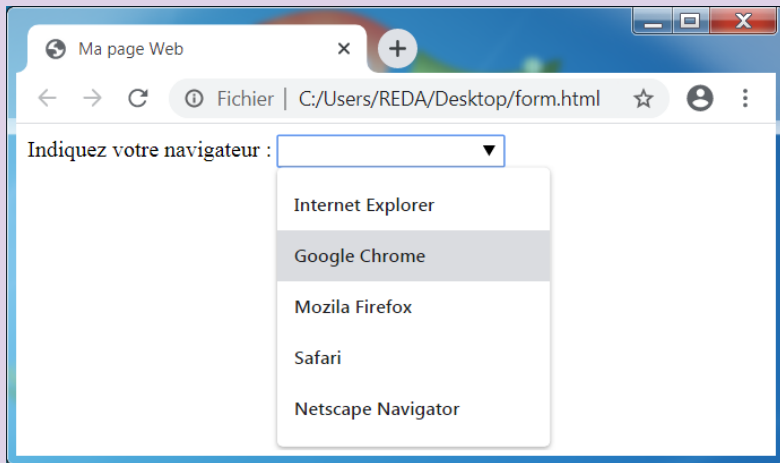
Ajout d'un champ de saisie/sélection

Quand on clique à l'intérieur du champ on obtient :



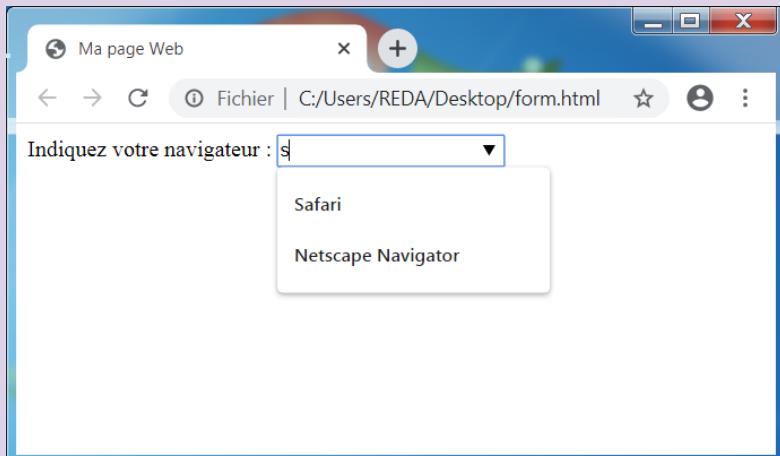
Ajout d'un champ de saisie/sélection

On peut donc choisir une valeur prédéfinie, comme par exemple :



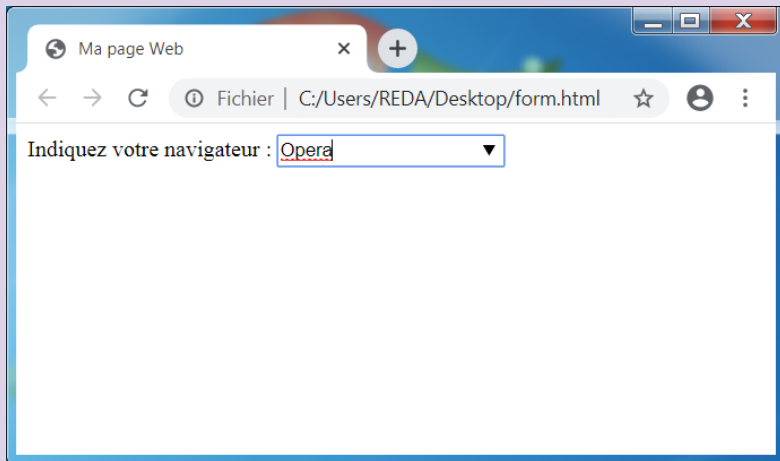
Ajout d'un champ de saisie/sélection

On peut limiter les choix à par exemple uniquement les valeurs prédéfinies qui contiennent la lettre s :



Ajout d'un champ de saisie/sélection

Ou on peut saisir une nouvelle valeur comme par exemple :



Ajout d'un champ de saisie/sélection

- L'attribut *list* de la balise `<input />` permet d'indiquer l'identifiant de la liste des valeurs qui va être utilisée.
Cet identifiant va être attribuer à l'attribut *id* de la balise `<datalist>`.
- L'attribut *name* de la balise `<input />` permet de spécifier le nom de la variable qui sera envoyé.
- Chaque couple `<option>` et `</option>` indique une valeur de la liste.
- Remarque : La valeur envoyée sera soit une valeur prédéfinie sélectionnée ou soit une nouvelle valeur saisie.

Ajout d'un champ de mot de passe

- Un champ de mot de passe ressemble à un champ texte sur une seule ligne, à la différence que les caractères qu'on y saisi sont masqués.
- Nous allons utiliser la balise `<input />` avec le couple attribut valeur `type="password"`.
- L'attribut *name* de la balise `<input />` permet de spécifier le nom de la variable qui sera envoyé.
- L'attribut *value* de la balise `<input />` permet de donner une valeur par défaut à la variable.
- Exemple :
Entrez votre mot de passe : `
`
`<input type="password" name="motdepasse" value="xxxxx" />`

Ajout d'une case à cocher

- Nous allons utiliser la balise `<input />` avec le couple attribut valeur `type="checkbox"`.
- L'attribut *name* de la balise `<input />` permet de spécifier le nom de la variable qui sera envoyé.
- L'attribut *value* de la balise `<input />` permet de spécifier la valeur qui sera envoyée si la case est cochée.
- Nous pouvons afficher une case pré-cochée en ajoutant le couple attribut valeur `checked="checked"` dans la balise `<input />`.

- Exemple :

Catégorie socio-professionnelle : `
`

`<input type="checkbox" name="socioprof1" value="etudiant" checked="checked"/>` Étudiant

`<input type="checkbox" name="socioprof2" value="chomeur"/>` En recherche d'emploi

`<input type="checkbox" name="socioprof3" value="fonctionnaire"/>` Fonctionnaire

`<input type="checkbox" name="socioprof4" value="retraite"/>` Retraité

Ajout d'un bouton d'option (bouton radio)

- Nous allons utiliser la balise `<input />` avec le couple attribut valeur `type="radio"`.
- L'attribut *name* de la balise `<input />` permet de spécifier le nom de la variable qui sera envoyé.
- L'attribut *value* de la balise `<input />` permet de spécifier la valeur qui sera envoyée si l'option est sélectionnée.
- Nous pouvons afficher un bouton radio pré-coché en ajoutant le couple attribut valeur `checked="checked"`.
- Remarque : Les boutons radio se distinguent des cases à cocher :
 - – Visuellement, par leur forme ronde,
 - – fonctionnellement, par une différence de comportement. Les boutons radios n'autorisent qu'un seul choix, là où les checkboxes permettent de cocher plusieurs cases.

Ajout d'un bouton d'option (bouton radio)

● Exemple :

Civilité : `
``<input type="radio" name="civile" value="Mme"/>` Madame`<input type="radio" name="civile" value="Mlle"/>` Mademoiselle`<input type="radio" name="civile" value="Mr"/>` Monsieur

Ajout d'un bouton d'envoi (bouton submit)

- Nous allons utiliser la balise `<input />` avec le couple attribut valeur `type="submit"`.
- L'attribut *value* de la balise `<input />` permet d'indiquer le texte qui sera affiché dans le bouton.
- Lors du clic sur ce bouton, les données du formulaire seront envoyées à la page Web précisée dans l'attribut *action* de la balise `<form>`.
- Exemple :
`<input type="submit" value="Envoyer"/>`

Ajout d'un bouton d'annulation (bouton reset)

- Nous allons utiliser la balise `<input />` avec le couple attribut valeur `type="reset"`.
- L'attribut *value* de la balise `<input />` permet d'indiquer le texte qui sera affiché dans le bouton.
- Lors du clic sur ce bouton, les données saisies et/ou choisies par l'utilisateur seront effacées ; il y aura ainsi restitution des valeurs par défaut.
- Exemple :
`<input type="reset" value="Annuler"/>`

Création et ajout d'un bouton dit "simple"

- Nous allons utiliser la balise `<input />` avec le couple attribut valeur `type="button"`.
- L'attribut *value* de la balise `<input />` permet d'indiquer le texte qui sera affiché dans le bouton.
- Lors du clic sur ce bouton, aucune action ne sera effectuée. Néanmoins, on pourra lui attribuer une action avec du code JavaScript.

Les attributs *placeholder* et *pattern* de la balise `<input />`

- La balise `<input />` a un attribut *placeholder* qui permet d'afficher dans le champ (avant que l'utilisateur n'entame la saisie) un exemple ou une description du format de la valeur attendue dans ce champ.
- La balise `<input />` a un attribut *pattern* qui permet de définir une expression régulière que la valeur saisie dans le champ devra respecter.
- Exemples :

Le texte saisi doit commencer par une ou deux lettre(s) majuscule(s), puis contenir un chiffre de 1 à 9, et finir avec cinq chiffres de 0 à 9 chacun.

```
<input type="text" name="cin" placeholder="A123456" pattern="[A-Z]{1,2}[1-9][0-9]{5}"/>
```

Le texte saisi doit contenir douze chiffres de 0 à 9 chacun, tout en espaçant les séquences de quatre chiffres par un caractère -.

```
<input type="text" name="numero-carte" placeholder="0123-4567-8910" pattern="[0-9]{4}-[0-9]{4}-[0-9]{4}"/>
```

D'autres attributs de la balise `<input />`

- La balise `<input />` admet le couple attribut valeur *required*="required" qui permet de rendre le champ en question obligatoire.
- La balise `<input />` a un attribut *min* qui permet de définir la valeur minimale que pourra prendre l'élément. *min* sera soit un nombre ou soit une date.
- La balise `<input />` a un attribut *max* qui permet de définir la valeur maximale que pourra prendre l'élément. *max* sera soit un nombre ou soit une date.
- La balise `<input />` admet le couple attribut valeur *size*="N" qui permet de définir la largeur d'un champ de saisie. Notez bien qu'on peut introduire dans le champ un nombre de caractères plus élevé que la valeur N.
- La balise `<input />` admet le couple attribut valeur *maxlength*="N" qui permet de limiter le nombre réel de caractères qu'on peut introduire dans le champ de saisie à uniquement la valeur N.

Remarques

- La balise `<input />` permet d'ajouter d'autres types d'éléments dans les formulaires :
 - `<input type="date" name="nom-variable"/>` ajoute un champ pour saisir une date ou la sélectionner dans un calendrier qui s'affiche,
 - `<input type="time" name="nom-variable"/>` ajoute un champ pour saisir ou choisir une heure,
 - `<input type="file" name="nom-variable"/>` ajoute un bouton pour parcourir l'arborescence de la machine pour choisir un fichier,
 - `<input type="email" name="nom-variable"/>` ajoute un champ pour saisir une adresse e-mail. Notez bien que les données saisies dans le formulaire ne seront envoyées que si l'adresse e-mail saisie est de la forme `texte@texte`,
 - `<input type="number" name="nom-variable"/>` ajoute un champ qui ne permet de saisir qu'une valeur entière,
 - ...

Remarques

- Nous avons vu jusqu'à maintenant des champs de formulaire qui s'obtiennent avec la balise `<input />`. Nous allons voir dans ce qui suit des champs de saisie qui utilisent des balises spécifiques. Notez bien que ces dernières vont s'insérer elles aussi entre les balises `<form>` et `</form>`.

Ajout d'un champ texte sur plusieurs lignes (zone de texte)

- Nous allons utiliser les balises `<textarea>` et `</textarea>`.
- L'attribut *name* de la balise `<textarea>` permet de spécifier le nom de la variable qui sera envoyé.
- L'attribut *rows* de la balise `<textarea>` permet de préciser le nombre initial de lignes de la zone de texte (qu'on pourra dépasser après lors de la saisie). Plus le chiffre est grand, plus la textarea sera plus longue.
- L'attribut *cols* de la balise `<textarea>` permet de préciser le nombre de caractères par ligne. Plus le chiffre est grand, plus la textarea sera plus large.
- Un texte placé entre les balises `<textarea>` et `</textarea>` permettra de donner une valeur par défaut à la variable.
- La balise `<textarea>` admet le couple attribut valeur *required*="required" qui permet de rendre le champ obligatoire.

Ajout d'un champ texte sur plusieurs lignes (zone de texte)

- Exemple :

Commentaire : `
`

`<textarea name="commentaire" rows="6" cols="40">` Veuillez nous
laisser un commentaire. `</textarea>`

Ajout d'un champ de sélection

- Les champs de sélection permettent de proposer une liste déroulante de choix.
- Nous allons utiliser les balises `<select>` et `</select>` pour créer une liste déroulante de choix.
- Entre ces dernières, nous allons placer autant de couple `<option>` et `</option>` que nous avons de choix.
- L'attribut *name* de la balise `<select>` permet de spécifier le nom de la variable qui sera envoyé.
- La balise `<select>` admet le couple attribut valeur *required*="required" qui permet de rendre obligatoire la sélection d'une valeur.
- En mettant le couple attribut valeur *size*="N" dans la balise `<select>` on détermine le nombre d'éléments de la liste qui sont affichés en même temps.
- Pour chaque balise `<option>`, nous précisons la valeur qui sera envoyée avec l'attribut *value*.

Ajout d'un champ de sélection

- Nous précisons également le texte qui sera affiché pour le visiteur, en le plaçant entre les balises `<option>` et `</option>`.
- Nous pouvons mettre en place un choix par défaut via l'ajout de `selected="selected"` dans la balise `<option>` en question.
- Exemple :

Catégorie socio-professionnelle : `
`

```
<select name="socioprof">
```

```
  <option value="chomeur">En recherche d'emploi</option>
```

```
  <option value="etudiant" selected="selected">Étudiant</option>
```

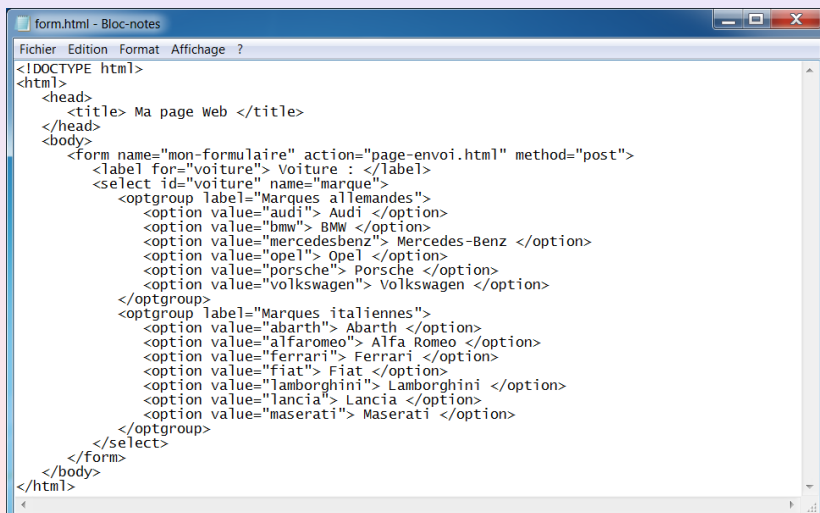
```
  <option value="fonctionnaire">Fonctionnaire</option>
```

```
  <option value="retraite">Retraité</option>
```

```
</select>
```

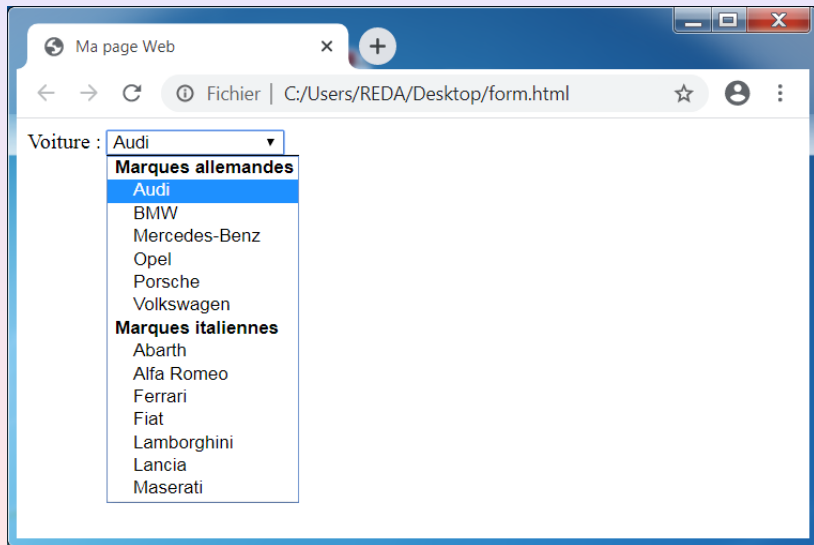
- Remarque : On peut regrouper des éléments d'une liste déroulante en un groupe nommé, en délimitant ces éléments par des balises `<optgroup>` et `</optgroup>`.
- Exemple :

Ajout d'un champ de sélection



```
Fichier Edition Format Affichage ?
<!DOCTYPE html>
<html>
  <head>
    <title> Ma page Web </title>
  </head>
  <body>
    <form name="mon-formulaire" action="page-envoi.html" method="post">
      <label for="voiture"> Voiture : </label>
      <select id="voiture" name="marque">
        <optgroup label="Marques allemandes">
          <option value="audi"> Audi </option>
          <option value="bmw"> BMW </option>
          <option value="mercedesbenz"> Mercedes-Benz </option>
          <option value="opel"> Opel </option>
          <option value="porsche"> Porsche </option>
          <option value="volkswagen"> Volkswagen </option>
        </optgroup>
        <optgroup label="Marques italiennes">
          <option value="abarth"> Abarth </option>
          <option value="alfaromeo"> Alfa Romeo </option>
          <option value="ferrari"> Ferrari </option>
          <option value="fiat"> Fiat </option>
          <option value="lamborghini"> Lamborghini </option>
          <option value="lancia"> Lancia </option>
          <option value="maserati"> Maserati </option>
        </optgroup>
      </select>
    </form>
  </body>
</html>
```

Ajout d'un champ de sélection



Les balises "label"

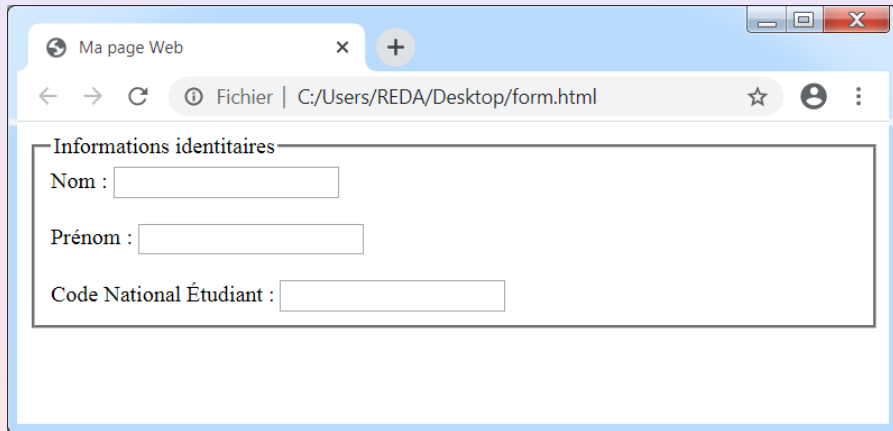
- Remarque : Les balises `<label>` et `</label>` peuvent être utilisées pour associer et afficher un libellé à côté d'un élément d'un formulaire.
- En plus de l'utilisation avec la balise `<select>`, notez bien qu'elles peuvent être utilisées aussi avec une balise `<input />` et avec une balise `<textarea>`.
- La valeur de l'attribut *for* de la balise `<label>` doit être égale à la valeur de l'attribut *id* qu'on ajoutera dans la balise `<select>` ou `<input />` ou `<textarea>`.
- Sachez qu'en cliquant sur un libellé, l'élément du formulaire auquel il est associé se voit activé, i.e., qu'on peut commencer à écrire/sélectionner dedans, ce qui peut offrir donc une meilleure ergonomie d'utilisation sur les écrans à petite taille.

Les balises "fieldset" et "legend"

- Les balises `<fieldset>` et `</fieldset>` permettent de regrouper des éléments d'un formulaire.
- Les balises `<legend>` et `</legend>` permettent d'ajouter une légende (un titre) au groupe d'éléments défini par les balises "fieldset".
- Exemple :

```
<fieldset>
  <legend> Informations identitaires </legend>
  <label for="nom"> Nom : </label>
  <input id="nom" type="text" name="nom"/>
  <br /> <br />
  <label for="prenom"> Prénom : </label>
  <input id="prenom" type="text" name="prenom"/>
  <br /> <br />
  <label for="cne"> Code National Étudiant : </label>
  <input id="cne" type="text" name="cne"/>
</fieldset>
```

Les balises "fieldset" et "legend"



The screenshot shows a web browser window with a single tab titled 'Ma page Web'. The address bar shows the file path 'C:/Users/REDA/Desktop/form.html'. The page content is a form titled 'Informations identitaires' which is enclosed in a `fieldset` element. Inside the fieldset, there are three input fields: 'Nom :', 'Prénom :', and 'Code National Étudiant :'. The browser's status bar at the bottom shows various navigation icons.

Ma page Web

← → ↻ ⓘ Fichier | C:/Users/REDA/Desktop/form.html ☆ ⓘ

Informations identitaires

Nom :

Prénom :

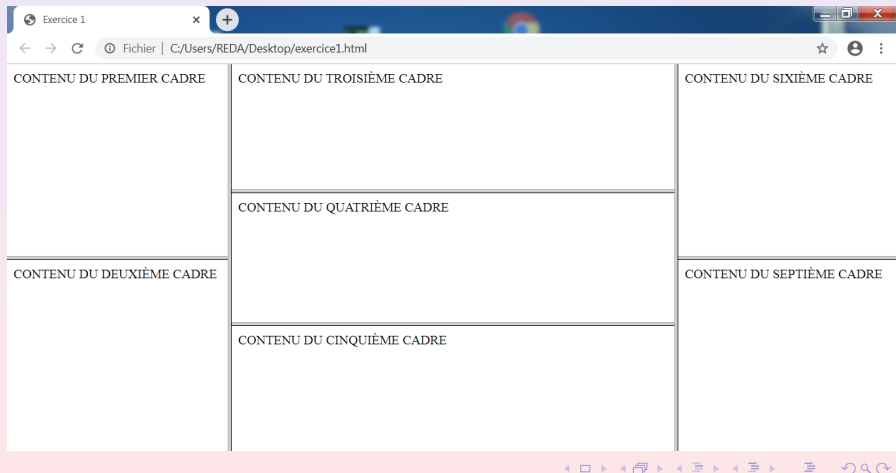
Code National Étudiant :

Conclusion

- HTML est un langage qui englobe une multitude de balises qui permettent de créer des pages Web statiques.
- HTML permet de mettre en page et de structurer du texte, d'avoir des liens hypertextes, d'ajouter du contenu multimédia, d'utiliser les frames, et de créer des formulaires.
- Remarque : On peut afficher le code source d'une page Web donnée, en cliquant sur le bouton droit de la souris au niveau de la page Web (au niveau du navigateur), et en sélectionnant l'action "Afficher le code source de la page".

Exercice 1

Créez le site Web suivant :



Exercice 2

- Créez une page Web qui contient un formulaire qui permettra aux visiteurs de saisir toutes les informations qu'on met habituellement dans un curriculum vitæ.